EFFECTIVE DATE November 5, 1996	LANL-YMP-QP-03.21, R7 Page 1 of 21

SOFTWARE LIFE CYCLE

LOS ALAMOS
QUALITY PROGRAM

APPROVAL FOR RELEASE

B. D. GUNDLACH - PREPARER Signature on file	DATE Date on file
M. J. CLEVENGER - QUALITY ASSURANCE PROJECT LEADER Signature on file	DATE Date on file
J. A. CANEPA - TECHNICAL PROJECT OFFICER Signature on file	DATE Date on file

Los Alamos
Yucca Mountain Site
Characterization Project

HISTORY OF REVISION

REVISION NO.	EFFECTIVE DATE	PAGES REVISED	REASON FOR CHANGE
R0	01/25/91	N/A	Initial procedure.
R1	01/31/94	All	Complete rewrite. Simplify process and incorporate QARD requirements.
R2	08/01/94	3, 4, 8-11, & 14	To incorporate RTN review comments and to specify personnel training.
R3	11/03/94	7 & 9-11	To incorporate additional RTN review comments.
R4	09/28/95	All	Complete rewrite to represent the flow of the developer and configuration manager in a simplified in-line approach.
R5	12/21/95	3, 4, 6, 10, 14, 26, 27, 36, 37, 39, & 47	Changes made to remain concurrent with revised QARD and to implement paperwork reduction documentation changes.
R6	08/08/96	All	Editorial and format changes to enhance readability.
R7	11/05/96	4 & 8	Minor, non-substantive change in response to DOE audit concern. Added name and version of software-requirement to subsection 6.1. Deleted NOTE: from pg.8.

SOFTWARE LIFE CYCLE

1.0 PURPOSE

This procedure describes the life cycle of and process for obtaining and developing new or maintaining existing Los Alamos National Laboratory (Los Alamos) Yucca Mountain Site Characterization Project (YMP or Project) software. The format of this procedure includes in-line documentation examples instead of separate attachments.

2.0 SCOPE

2.1 This procedure governs software developed or acquired for use in support of licensing activities.

2.2 This procedure applies to all Los Alamos and Los Alamos-subcontractor YMP personnel (hereafter referred to as YMP personnel) who work under the Los Alamos YMP quality assurance program.

3.0 REFERENCES

AP-16.1Q, Performance / Deficiency Reporting
AP-16.2Q, Corrective Action and Stop Work
ANSI / IEEE Standards 830-1984, IEEE Guide to Software Requirements Specifications, July 1984
File List Standards (ECD-9)
LANL-YMP-QP-03.5, Documenting Scientific Investigations
LANL-YMP-QP-03.20, Software Configuration Management
LANL-YMP-QP-17.6, Records Management

4.0 DEFINITIONS

4.1 Software Heritage

The software heritage is a designation that specifies the previous development history of a software application. Two heritage designations are supported: acquired software (AS) and new development software (NDS).

5.0 RESPONSIBILITIES

The following YMP personnel are responsible for activities identified in Section 6.0 of this procedure:

- Software Change Originator (Originator)
- Software Configuration Manager (SCM)
- Configuration Control Board (CCB)
- Software Developer (Developer)
- Reviewers
- Analysts

6.0 PROCEDURE

The use of this procedure must be controlled as follows:

- If this procedure cannot be implemented as written, YMP personnel should notify appropriate supervision. If it is determined that a portion of the work cannot be accomplished as described in this QP, or would result in an undesirable situation, that portion of the work will be stopped and not resumed until this procedure is modified or replaced by a new document that reflects the current work practice.

- YMP personnel may use copies of this procedure printed from the controlled document electronic file; however, YMP personnel are responsible for assuring that the correct revision of this procedure is used.

- When this procedure becomes obsolete or superseded, it must be destroyed or marked “superseded” to ensure that this document is not used to perform work.

6.1 Software Qualification.

Commercial or government-off-the shelf software packages (e.g., EXCEL, Lotus 1-2-3, @Risk, spreadsheets, Microsoft Access, dBase, FoxPro, database managers, DOS, Windows, operating systems, administrative and management systems, system compilers, utilities, and associated libraries, WORD, WordPerfect, word processing programs, graphing and visual display software, statistical analysis programs, or software included with test or data gathering) are exempt from qualification even when quality affecting formulas (or user developed macros) are included in fields or cells for calculations. The formulas (or user developed macros) themselves will be documented, verified and validated (e.g., peer review) if used to support quality affecting work. Include a list of the formulas or macros and software name and version (e.g., EXCEL 4.0 or Microsoft Access 2.0) in your notebook or report you publish describing quality affecting work. The intent is to enable a person knowledgeable in your subject to repeat your work using the information provided in your notebook or report. Refer to QP-03.5 for additional information.

6.2 Software Development, Modification and Certification.

YMP personnel fill out a Software Change (SC) form, (Attachment 1 and 2). The SC form provides a means for proposing enhancements to existing software or for requesting authorization to develop new software to comply with the provisions of the Los Alamos YMP software quality assurance program. The SC is also used to initiate the qualification of AS.

6.2.1 Submission of a SC.

Before the initiation of the formal software life-cycle for the development of the first version of a software application.

- To initiate the qualification of AS not controlled by another Yucca Mountain Project Software Configuration Management Office (YMP/SCMO).

- During the Design, or Implementation Phase to request CCB approval to revisit a prior (closed) life cycle phase.

- During the Requirements, Design, or Implementation Phase to document formally an enhancement suggested for incorporation later, outside of the scope of the current development activity.

- To report a problem or discrepancy with a Los Alamos SCM controlled software application or baseline.

- To initiate withdrawal and retirement of baseline elements.

6.2.2 If YMP personnel are identifying a problem with the software, use the following criteria as part of your description.

- Does the problem occur throughout the entire system (or software suite), happen with a particular application or program, pertain to the software's documentation, or is this a data problem.

- Does the problem involve:

  a. A data reference,

  b. Declaration of data,

  c. Computation error,

  d. Comparison error,

  e. Program control flow,

  f. Interface (i.e., wrong disk, file, display),

  g. Input/output, or

  h. Other - describe the type in the analysis block.

- Were you able to recover from the problem?

- Is the condition repeatable?

- For changes requested due to a Performance, Deficiency, or Corrective Action Report (PR, DR, or CAR), give the PR, DR, or CAR report number as appropriate.

6.3 Determination of a SC Disposition.

If the SC was not returned by the SCM for “Insufficient Information” (Insufficient Information box checked in the DISPOSITION section), then the SC is forwarded to the CCB Chairperson. The **CCB Chairperson** will contact the CCB members and if necessary hold a meeting to discuss the SC. A formal meeting is not necessary, the important part is that the **CCB members** participate in the evaluation and decision process.

6.3.1 The **CCB** evaluates the proposed SC to determine the disposition of the SC. This is accomplished by:

- The **SCM** provides discrepancy-analysis support to the CCB and analyst by utilizing the Configuration Status Accounting (CSA) records and the SCM controlled datasets to provide support to the CCB and analyst in determining the cause, impact on previous calculations, and possible corrective actions for discrepancies reported on an SC.

- The **SCM** generates on-demand CSA reports to establish the scope of discrepancy impact and supply the mechanism for notifying affected users of suspect software. Affected users and organizations will be appropriately notified by either electronic mail message and/or mailed written notice relaying "heads-up" information associated with the continued use of the software. Information will include, at a minimum, the name and version of the software and a summary of the reported discrepancy.

- If sufficient expertise exists on the CCB to determine the nature, impact, and scope of the proposed change or problem, the **CCB** assigns itself as analyst and performs the analysis of the SC.

- If there is insufficient expertise on the CCB to make this determination, the **CCB** assigns an analyst to perform additional evaluation.

6.3.2 The CCB’s prescribed analysis of the SC is performed as follows:

- If an existing software application baseline is affected by the SC, the **analyst** obtains an assessment release from the SCM by preparing and submitting a Software Transmittal (ST) form, (Attachments 3 and 4).

6.3.3 The **analyst** then performs the assessment as follows:

- Evaluates the scope and nature of the SC.

- Identifies the affected baseline elements on the SC.

- Assesses the impact of the proposed changes or discrepancies on design, other baselines, or applications.

- Determines whether the discrepancy affects the results of previous calculations.

- Establishes whether the discrepancy compromises results published in any formal reports.

6.3.4 The **analyst** next formulates a plan of corrective action to eliminate the discrepancy and mitigate any problems with affected software baselines or formal reports. Then the analyst documents the results of the analysis and proposed corrective action plan in the ANALYSIS section of the SC form.

6.3.5 For problems that adversely affect previous applications, the **analyst** will initiate a report within the framework of AP-16.1Q or AP-16.2Q, and annotate the CAR reference line of the PROBLEM CHARACTERIZATION section with the identifying number of the resulting PR, DR, or CAR.

6.3.6 Finally, the **analyst** signs and dates the ANALYST line at the bottom of the form and returns it to the SCM.

6.3.7 The **CCB** determines the disposition of the SC as follows:

6.3.7.1 Insufficient Information. Check this box if there was insufficient information to analyze the SC. The **CCB Chairperson** prints his or her name on, signs and dates the CCB Chair line on the form, and returns it to SCM.

6.3.7.2 Not Approved. If the request is rejected by the CCB, check this box. The **CCB Chairperson** prints his or her name on, signs and dates the CCB Chair line on the form, and returns it to SCM.

6.3.7.3 Approved. If the CCB approves the request, check the appropriate box. The **CCB Chairperson** prints his or her name on, signs and dates the CCB Chair line on the form, and returns it to SCM.

6.4 Determination of reviews, baselines, and documentation.

For acquired *certified* software *obtained from and controlled by another* YMP/SCMO, used for the same purpose and input/output range, and validated by successfully accomplishing the software's installation and checkout procedures), the **releasing organization** determines the reviews, baselines and documentation requirements. Control this software and associated documentation according to the procedures of the releasing organization.

For new development and non-YMP/SCMO AS, the **CCB** determines documentation and review requirements based on the software heritage. Following approval and assignment of an SC, the **CCB** documents and approves a specific life cycle plan for each software item prior to development or modification of software or the qualification of AS. The **CCB** documents the life cycle plan on the Life Cycle (LC) form, Attachment 7.

6.4.1 The **CCB** prepares the LC form (Attachments 7 and 8) to document necessary baselines, documentation, and reviews as described in the following paragraphs.

6.4.1.1 The **CCB** identifies the life-cycle phases that will be traversed for the development task. The areas to be addressed on the LC form are as follows:

- New Development: Requirements, Design, Implementation.
- Acquired Non-YMP/SCMO: Requirements, Implementation.
- For maintenance and problem or deficiency correction activities, the software life-cycle is specified based upon which documentation components of the subject application are impacted by the enhancement or repair task.
- To validate software for use outside the certified baseline's range of validation, the software life-cycle includes, at the minimum, re-accomplishment of the software's Implementation Baseline revising the V&V Plans and Procedures, the Test Results (TR), V&V Report, and User's Manual.

6.4.1.2 The **CCB** identifies the documentation components that will be developed for the life cycle phases and allocate each component to the appropriate life-cycle phase by checking the corresponding boxes on the LC form. Allocate each documentation component into one (and only one) of the baselines **associated with the software heritage described below.**

a**.** All new development software (NDS) includes Requirements, Design, and Implementation Phases. It also includes the following documentation:

- Methods and Models Summary (MMS).

NOTE: The MMS may be omitted if the model can be cited in technical literature. This citation can be made in the Software Requirements Specification.

- Software Design Document (SDD) (Attachment 9).
- Software Requirements Specification (SRS).
- TR.
- User's Manual / User's Guide (UM / UG).

  NOTE: For Reuse-Component Baselines only, the CCB may allow the UM to follow the alternative UG format.

- Version Description Document (VDD).
- Verification and Validation Plan and Procedures (VVP).

NOTE: The VVP is generally produced during the Implementation Phase. However the CCB may require that the plan portion of the VVP be produced during the Design Phase.

- Verification and Validation Report (VVR).

b. Acquired Non-YMP / SCMO Software (AS) includes Requirements and Implementation Baselines and the production of the following documentation:

- SRS.

NOTE: The SRS only needs to detail the requirements for which the software was acquired.

- UM / UG.
- VDD.
- VVP.

NOTE: The VVP is generally produced during the Implementation Phase. However the CCB may require that the plan portion of the VVP be produced during the Design Phase.

- TR.
- VVR.

c. Qualification of AS includes the following:

- Installation testing to ensure that software performs as required in the operational environment;

  Step 1 - Install the software as described in the System Interface section, Installation Instructions subsection of the software's UM.

  Step 2 - Start the software. Software should start without errors.

  Step 3 - Run one of the sample inputs described in the Examples and Sample Problems section of the software's UM. Software should complete without errors producing described result or output. If errors occur during any of the above steps, the installation failed. Resolve the error conditions before attempting another test.

- Confirmation that documented information exists to support that appropriate requirements were met;

  Step 1 - Inspect the requirements traceability section of the documentation prolog at the beginning of each of the software's modified or created main code modules.

  Step 2 - Compare the requirements traceability reference to the information in the software's SRS. If errors are found during any of the above steps, the confirmation failed. Resolve the errors before proceeding.

- Placing the software under configuration management. Refer to QP-03.20 for SCM procedures.

6.4.1.3 The **CCB** identifies the reviews that will be performed to certify the products of the development or deficiency repair task. Life cycle phases will be performed in this order -- Requirements, Design, and Implementation. If the Requirements, Design, and/or Implementation baselines are specified, select the corresponding baseline-closure reviews (Software Requirements Review (SRR), Critical Design Review (CDR), and Software Acceptance Review (SAR), respectively) by checking the appropriate box(es) on the LC form (Attachments 7 and 8). Optionally, an in-process review may be specified for any or all of these baselines by checking the corresponding In-Process Review box(es).

6.4.1.4 The sequence that the **CCB** will use to perform the reviews is as follows:

For example: The CCB could sequence all the In-Process reviews prior to sequencing the baseline-closure reviews.

Requirements In-Process sequence 1
Design In-Process sequence 2
Implementation In-Process sequence 3
SRR sequence 4
CDR sequence 5
SAR sequence 6
Or follow an in-line sequence.

Requirements In-Process sequence 1
SRR sequence 2
Design In-Process sequence 3
CDR sequence 4
Implementation In-Process sequence 5
SAR sequence 6.

6.4.1.5 If the Preliminary Design, V&V Plan, and / or V&V Procedures baselines are specified, select the corresponding in-process reviews by checking the appropriate box(es) on the LC form. Specify a sequence for each review in the space provided on the form.

6.4.1.6 Appropriate baselines are completed and audited by the SCM in the sequence specified on the LC form. Records of these completed reviews are available from the SCM. Refer to QP-03.20 for additional information.

6.4.2 During software development, it may be necessary to perform a rapid prototyping activity. Rapid prototyping is a process where the Developer suspends the current development activity and investigates, informally, alternative design approaches, algorithms, or other issues critical to the specification, design or implementation of the software application. Rapid prototyping permits tasks (e.g., coding) to be performed that would normally take place during a future development phase**.**

6.4.3 Rapid prototyping is not a substitute for the formal life-cycle-based development cycle, or any phase thereof. A rapid prototyping activity suspends the normal development life-cycle. The results of the informal investigation are then employed to guide the subsequent formal development efforts.

6.4.4 The **CCB Chairperson** authorizes the life cycle by signing and dating the CCB CHAIR line at the bottom of the LC form, and returning it to the SCM.

6.5 Beginning the development task.

The **Developer** may begin the software development task upon issuance of the LC from the CCB.

6.5.1 If changes will be made to any certified baseline to implement the work authorized on the SC, the **Developer** initiates an engineering release by preparing a ST form (Attachments 3 and 4). The developer than submits the form to the SCM and awaits receipt of the requested material.

6.5.2 For the traversal of the Software LC, proceed through the life-cycle phases in the sequence that they are specified on the LC form. Within each phase, the **Developer** develops the baselines and baseline components that are specified on the LC form. Unless to comply with a resubmission requested after a review, omit any phases, baselines, and baseline components that are not specified on the LC form. Employ a development strategy that effectively addresses the relevant technical issues and produces baseline components that comply with the following instructions. Before preparing any proposed baseline, the **Developer** confirms that the phase for this baseline is open. Confirmation may be obtained from the Software Configuration Manager. If the phase is not open, submit a SC to initiate a formal change processing cycle.

NOTE: All activities conducted within the scope of a life-cycle phase are directed toward:

- Generation of the baseline documents specified on the associated LC form for the phase.
- Completion of all reviews specified for the life-cycle phase on the associated LC form and resolution of all review / audit issues to the satisfaction of the Change Control Authority.

6.5.2.1 Components that make up a requirements baseline are as follows:

a. File List.

The **Developer** ensures that there is an entry for the SRS in the file list. If no file list exists, create the file list. Otherwise, update the existing file list.

Example : radprog.srs

b. SRS Format

The SRS documents the functional and performance requirements of the application and describes external interfaces and design constraints.

The **Developer** specifies software requirements which allows them to be easily identified and tracked through subsequent baselines. Ensure that the SRS clearly states and uniquely identifies each software requirement.

Emphasize the development of testable requirements. A clearly stated requirement can be easily identified in the Design or Implementation Baseline -- and specifically and objectively tested.

c. An example of the SRS standard format can be found in Attachment 9. The following example is the standard format of the SRS. Follow the brief description given for the contents of each major section. Parenthetical references in the section headings indicate sections in ANSI/IEEE Std 830-1984, *IEEE Guide to Software Requirements Specifications* from which additional description may be obtained.

6.5.2.2 The components of a design baseline are as follows:

- A file list.
- A SDD, if the LC Specification for the development task specifies it as a baseline component.
- For preliminary design - Primary source-code modules for the top-level and executive routines of each of the application's products.
- For detailed design - In addition to the primary source-code modules, identify each routine associated with the application and an MMS if the LC Specification for the development task specifies it as a baseline component.
- The V&V plan portion of the V&V Plan and Procedures document, if the Life Cycle Specification for the development task specifies it as a baseline component.

6.5.2.3 Create a file list that summarizes the components of the proposed baseline -- including any modules generated or changed for the baseline. The file list will match the components specified on the LC form.

6.5.2.4 The SDD is a development document that describes the top-level design of a software application.

6.5.2.5 An example of the SDD standard format can be found in Attachment 10. Follow the brief description given for the contents of each major section.

6.5.2.6 Documentation requirements for source code modules are as follows:

- The **Developer** includes the software version or revision identifier with generated output (for example in the header of a report), when feasible.
- During the Design Phase, the **Developer** documents traceability of the software requirements into the software design as follows.
  1. For the preliminary design, establish that major requirements (or groups of requirements) are allocated into the SDD, thereby demonstrating that the overall architecture of the software is adequate to accommodate the requirements specified in the SRS. For the detailed design, map each SRS requirement into one or more source code modules.
  2. Specify (in the Requirements Traceability section of each source-code documentation-prolog) the identifying number of each software requirement satisfied by the module. Ensure that each requirement in the SRS is implemented within one or more modules of the software design (as embodied in the documentation prolog of each software module).
- Each top-level primary source-code module will include a source-code documentation prolog.

6.5.2.7 An example of the standard source-code documentation prolog format can be found in Attachment 11. Follow the brief description given for the contents of each major section.

6.5.2.8 Format for a MMS.

The MMS is provided to apprise potential users of the details of the mathematical models and numerical methods employed by the software.

The following sections describe the contents of each major section of the MMS. In the event that any of the information required below is already present in an existing document such as a milestone report or journal article, the external document may be referenced and the information need not be reproduced here.

6.5.2.9 Follow the brief description given for the contents of each major section. An example of the standard MMS format can be located in Attachment 12.

6.5.2.10 Components that make up a V&V baseline are as follows:

- A file list.

  For V&V plan baseline, the **Developer** prepares the V&V plan portion of the V&V Plan and Procedures document. Prepares Sections 1.0 through 4.0 of the V&V Plan and Procedures as independent text-documents, as in-line documentation to the testing software, or as a combination of each as appropriate to the development task. Prepares the portions of the V&V Plan and Procedures that describe the program of formal testing (subsection 4.1) within the documentation prologs of the source-code modules that implement the testing software.

- For V&V procedures baseline, the **Developer**:

  a. If no V&V plan baseline exists, prepares the V&V plan portion of the V&V Plan and Procedures document. Prepares Sections 1.0 through 4.0 of the V&V Plan and Procedures as independent text-documents, as in-line documentation to the testing software, or as a combination of each as appropriate to the development task. Prepares the portions of the V&V Plan and Procedures that describe the program of formal testing (subsection 4.1) within the documentation prologs of the source-code modules that implement the testing software.

  b. Otherwise, updates the existing V&V Plan and Procedures to reflect any new features of the V&V program.

  c. Produces the V&V procedures by developing source code that implements the testing program described in subsection 4.1 of the V&V Plan and Procedures.

  d. Develops all test software necessary to support the testing effort (source-code prologs are optional for test software, but may be required by the CCB or reviewers).

6.5.2.11 Creates a file list that summarizes the components of the proposed baseline -- including any modules generated or changed for the baseline. The file list will match the components specified on the LC form.

6.5.2.12 The V&V Plan and Procedures document is development documentation that encapsulates the V&V plan and the V&V procedures that are derived from the plan. The V&V plan components of this document may be rendered as text documents, documentation prologs of the test software, or a combination of each, depending upon the information that will be represented.

6.5.2.13 The V&V procedures are always implemented in command language within the particular V&V Plan and Procedures document component that contains the corresponding V&V plan documentation. Follow the brief description given for the contents of each major section. An example of the V&V Standard format can be found in Attachment 13.

6.5.2.14 Components of an Implementation baseline are as follows:

- File list.
- All primary source-code modules. Develop code that corresponds to the module's documentation prolog and the detailed design.
- If the V&V Plan and Procedures is specified for the development task, complete this document and any associated support modules.
- If the TR document is specified for the development task, generate it and incorporate it into the baseline.
- If the UM (or Users Guide for Reuse Components) is specified for the development task, create it.
- If the MMS is specified for the development task and not created in a previous baseline, create it.
- If the VDD is specified for the development task, complete Section 2.0 and subsection 3.1. Do not complete any other sections of the VDD.
- If the V&V Report is specified for the development task, create it.

6.5.2.15 Software will be verified and validated before it can be released or used for any processing related to quality affecting work. Testing is the primary method of software validation. Software validation of modifications to released software items will include regression testing to ensure that the unmodified portion still functions correctly. Testing ensures that all requirements, as stated in the SRS, are fulfilled.

6.5.2.16 An individual independent of the code development implements V&V as follows:

- Emphasizes the development of testable requirements.
- Establishes that the software requirements are traceable into the software implementation (code).
- Writes source code that is a faithful rendition of the detailed design.

6.5.2.17 Use the V&V plan developed for the software. Develop test cases, as appropriate, of the following test classes:

- Verification tests are designed to demonstrate the correct operation of the software with respect to the documented software requirements.
- Validation tests are employed to demonstrate the veracity of a mathematical or numerical model.
- Regression tests represent a subset of the overall test suite. Regression tests are employed to test a new software version without executing the entire test suite.
- Acceptance tests are employed to demonstrate the correct operation of the software after installation on another computer system and provide a sample problem or suitable benchmark to demonstrate successful installation.

6.5.2.18 Test Execution and Assessment process as follows:

- V&V Procedures - Derive the V&V procedures from the V&V plan.
- TR - Generate the TR by executing the V&V procedures.
- Assessment - Compare the TR to the success criteria identified for each test in the test plan.
- Resolution - The **Developer** corrects any errors discovered in the previous step. If additional tests are required, repeat all steps of this V&V planning section. If errors are discovered in the requirements, design, or implementation, re-enter the appropriate life-cycle phase to correct the problem. (This may require initiation of a formal change processing activity). Repeat this step until all tests are passed.
- Generates a TR document.
- The **Developer** updates the V&V Report.

6.5.2.19 The **Developer** creates a file list that summarizes the components of the proposed baseline, including any modules generated or changed for the baseline. The file list will match the components specified on the LC form.

6.5.2.20 The TR document is development documentation that contains a record of the execution of the V&V procedures, i.e., the TR contains the output generated when the test procedures are executed on the target computer. As such it constitutes the primary record of the results of the software testing activity.

The TR document may be distributed among many separate documents. Each test suite may have a separate TR document, or each test driver may update a common TR document. The organization of the TR document is left to the discretion of the developer, but it will be acceptable to the CCB.

6.5.2.21 The User documentation format is as follows:

User documentation takes two forms for Los Alamos YMP controlled software.

a. For Reuse Components, write or update the UG for Reuse Components (UG).

b. For all other software, write or update the UM.

The UG or UM provides a user of the software application with information necessary to install and run the application. It is one of the principal components of user documentation.

6.5.2.22 Follow the brief description given for the contents of each major section. An example of the user documentation format can be found in Attachments 14 and 15.

6.5.2.23 The VDD describes the content and capabilities of each delivered version of the system. As such, it is a primary user's documentation component.

6.5.2.24 Follow the brief description given for the contents of each major section. An example of the VDD standard format can be found in Attachment 16.

6.5.2.25 What is the format for a V&V Report (VVR)?

The V&V Report presents a critical analysis of the TR with respect to the expected results specified in the V&V Plan and Procedures. The V&V Report is development documentation. The individual implementing V&V, includes the results of the verification testing effort in the V&V Report and interpret the results to develop a recommendation regarding whether the results are adequate to support verifying the software. If, due to circumstances beyond the control of the Developer, the software cannot be adequately verified, provide a justification for this position in Section 6.0 of the V&V Report. Also in this section, document a proposed schedule for completion of the verification effort at a level commensurate with the extent to which the software will be used to support a license application.

6.5.2.26 Follow the brief description given for the contents of each major section. An example of the VVR standard format can be found in Attachment 17.

6.6 Closing a Life-Cycle Phase

6.6.1 The initiating event for the closure of a life-cycle phase is a baseline closure review. The procedure for closing a life-cycle phase is given below. Table I summarizes the baseline closure reviews as they apply to the life-cycle phases.

- The life cycle form defines which closure reviews apply to the project and the sequence to perform the reviews. Follow the sequence shown on the LC form.

- Prepare the baseline, submit it for review, and resolve all review issues to the satisfaction of the Change Control Authority.

When the software has passed the baseline closure review at the end of the life-cycle phase, certification of the baseline by the SCM closes the life-cycle phase.

Life Cycle Phase	**Baseline Closure Review**	**Review of**
Requirements	Software Requirements Review (SRR)	Requirements Baseline
Design	Critical Design Review (CDR)	Detailed Design Baseline
Implementation	Software Acceptance Review (SAR)	Implementation Baseline

Table I. Life-Cycle Phase Closure

6.6.2 At this point, the **Developer** has prepared the baseline (ensuring that all components specified for the baseline are present and complete) and finished all tasks required for the particular baseline for the life cycle phase being submitted for in-process review or closure.

- Use the ST form to document the baseline submission.

- Next, forward the form plus the prepared baseline package to the SCM. If network access is impractical or impossible, submit the proposed baseline package on magnetic or optical media.

6.6.3 The **Developer** prepares a ST form. (reference Attachments 3 and 5 for instructions.)

6.7 Release, or Register of Sanctioned Los Alamos YMP Controlled Software

6.7.1 To obtain the latest version of a Los Alamos YMP sanctioned software application, a release of information from the SCM controlled datasets, or to register use of a sanctioned application already available, execute the following steps.

- Formal Request for Software or Data.
- The **Originator** prepares a ST form (Attachments 3 and 6).
- The **Originator** submits the form to the Los Alamos YMP SCM.

6.7.2 Upon receipt of the distribution package from the SCM, perform the installation according to the installation instructions enclosed. Execute the acceptance tests provided with the distribution package and compare the results to the certified results provided. If discrepancies are discovered, enlist the aid of the SCM to identify the cause and resolve the problem. If the discrepancies cannot be resolved, submit a SC form detailing the problem or discrepancies.

7.0 RECORDS

As described in QP-03.20, after certification, the corresponding baselines and supplemental documentation are incorporated into record packages by the **SCM**. The supplemental documentation forms that are included in these record packages are listed in Section 10.0. Record packages are prepared and submitted by SCM in accordance with QP-17.6.

8.0 TRAINING

8.1 Prior to conducting work described in Section 6.0, the developer, originator, members of SCM and CCB, analyst, and reviewers require training to this procedure.

8.2 Training to this procedure is accomplished by “read only”.

9.0 ATTACHMENTS

Attachment 1:	Software Change (1 page)
Attachment 2:	Software Change Instructions (1 page)
Attachment 3:	Software Transmittal (1 page)
Attachment 4:	Software Transmittal Instructions for Engineering Release (1 page)
Attachment 5:	Software Transmittal Instructions for Baseline Closure (1page)
Attachment 6:	Software Transmittal Instructions for Formal Request for Software or Data (1 page)
Attachment 7:	Life Cycle (1 page)
Attachment 8:	Life Cycle Instructions (1 page)
Attachment 9:	Software Requirements Specification (3 pages)
Attachment 10:	Software Design Document (2 pages)
Attachment 11:	Source-Code Documentation-Prolog (1 page)
Attachment 12:	Models and Methods Summary (2 pages)
Attachment 13:	V&V Plan and Procedures (2 pages)
Attachment 14:	Users Guide for Reuse Components (1 page)
Attachment 15:	Users Manual (4 pages)
Attachment 16:	Version Description Document (1 page)
Attachment 17:	V&V Report (1 page)

SOFTWARE CHANGE

DATE ____________ TITLE: ____

ORIGINATOR: ________________________________ **SC -** ____________

ORGANIZATION: ______________________ PHONE: _________ MAIL STOP: _________

DESCRIPTION OF PROPOSAL:

JUSTIFICATION:

DISPOSITION:

☐ INSUFFICIENT INFORMATION

☐ NOT APPROVED

☐ APPROVED

ANALYSIS:

ORIGINATOR:		Signature	Date
ANALYST:	Print name	Signature	Date
CCB CHAIR:	Print name	Signature	Date

LANL-YMP-QP-03.21

Los Alamos
Yucca Mountain Site Characterization Project

SOFTWARE CHANGE INSTRUCTIONS

- Date - Enter today's date.
- Title - Provide a brief summary of the proposed change.
- Originator - your name.
- Enter the SC Number.
- Organization - your organization (i.e., EES-13).
- Telephone - your work phone number.
- Mail Stop - your mail stop (i.e., M321).
- Description of Problem or Proposed Software Change - Provide a detailed description of the problem or proposed change. If enhancements to existing software are being proposed, be specific in identifying the affected software (see subsection 6.2.3).
- Proposed SC Justification - Describe the merits of the proposed change.
- Assign a Disposition.
- Enter an analysis of the problem to be corrected.
- Originator - Sign and date the Originator line at the bottom of the form.
- Return the form to SCM.

Los Alamos
Yucca Mountain Site
Characterization Project

SOFTWARE TRANSMITTAL

DATE: ____________ TITLE: ______________________________

ORIGINATOR: ______________________________ **ST -** ____________

ORGANIZATION: ____________________ PHONE: ________ MAIL STOP: ________

☐ REQUIREMENTS	☐ V&V PROCEDURES	☐ IN-PROCESS	DISTRIBUTION REQUEST
☐ PRELIMENARY DESIGN	☐ IMPLEMENTATION		☐ YMP USE
☐ V&V PLAN	☐ DETAILED DESIGN	☐ BASELINE CLOSURE	☐ EXTERNAL USE

FILE TRANSMITTAL BLOCK

CONTACT: ____________ PHONE: ____________ MAIL STOP: ____________

HARDWARE PLATFORM: ____________ OPERATING SYSTEM: ____________

FILE FORMAT: ______________________________

METHOD OF DISTRIBUTION

☐ TAPE ☐ DISK ☐ NETWORK FILE TRANSFER

DETAILS:

☐ ACCEPTED	SCM/CCB COMMENTS:
☐ CCB ACTION	
☐ NOT ACCEPTED	

ORIGINATOR: ____________ ________
Signature Date

SCM REPRESENTATIVE: (Disposition) ____________ ____________ ________
Print name Signature Date

LANL-YMP-QP-03.21

Los Alamos
Yucca Mountain Site Characterization Project

SOFTWARE TRANSMITTAL INSTRUCTIONS FOR ENGINEERING RELEASE

1. Date - Today's date.
2. Title - Full application name or title of document.
3. Originator - Your name.
4. ST - enter the task SC number followed by a '-' followed by the task LC number. (SCM will fill in the last unique identifying number.).
5. Organization - Your organization (i.e., EES-13).
6. Phone -Your work phone number.
7. Mail Stop - If for Los Alamos YMP use, your mail stop. Otherwise put N/A.
8. Check the YMP USE box.

Instructions 9-14 are for the FILE TRANSMITTAL BLOCK.

9. Contact - Enter "ORIGINATOR" or name of person who will handle the transfer.
10. Phone - Contact work phone number.
11. Mail Stop - Contact mail stop, if not Originator and YMP use.
12. Hardware Platform - Destination platform (i.e., VAX, Sun SPARC, IBM compatible PC).
13. Operating System - Destination computing systems operating system (i.e., VAX VMS, UNIX, DOS, OS/2, IBM MVS).
14. File Format - Specify the preferred file format for transmission.

Instructions 15-16 are for the METHOD OF DISTRIBUTION BLOCK.

15. Method of Distribution - Specify the preferred method of transmission by checking one of the Tape, Disk, or Network File Transfer boxes.
16. Details - Provide the detailed information needed to make the transfer. (If for external use provide your full mailing, e-mail, or network address based on your transfer choice.).
17. Sign and date the ORIGINATOR line at the bottom of the form.

Los Alamos
Yucca Mountain Site Characterization Project

SOFTWARE TRANSMITTAL INSTRUCTIONS FOR BASELINE CLOSURE

1. Date - Today's date.
2. Title - Full application name or title of document.
3. Originator - Your name.
4. ST - enter the task SC number followed by a '-' followed by the task LC number. (SCM will fill in the last unique identifying number.).
5. Organization - Your organization (i.e., EES-13).
6. Phone -Your work phone number.
7. Mail Stop - If for Los Alamos YMP use, your mail stop. Otherwise put N/A.
8. Check the boxes corresponding to the baseline you are submitting. Consult the associated development or maintenance task LC form.
9. Check the YMP USE box.

Instructions 10-15 are for the FILE TRANSMITTAL BLOCK.

10. Contact - Enter "ORIGINATOR" or name of person who will handle the transfer.
11. Phone - Contact work phone number.
12. Mail Stop - Contact mail stop, if not Originator and YMP use.
13. Hardware Platform - Destination platform (i.e., VAX, Sun SPARC, IBM compatible PC).
14. Operating System - Destination computing systems operating system (i.e., VAX VMS, UNIX, DOS, OS/2, IBM MVS).
15. File Format - Specify the preferred file format for transmission.

Instructions 16-17 are for the METHOD OF DISTRIBUTION BLOCK.

16. Method of Distribution - Specify the preferred method of transmission by checking one of the Tape, Disk, or Network File Transfer boxes.
17. Details - Provide the detailed information needed to make the transfer. (If for external use provide your full mailing, e-mail, or network address based on your transfer choice.).
18. Sign and date the ORIGINATOR line at the bottom of the form.

Los Alamos
Yucca Mountain Site Characterization Project

SOFTWARE TRANSMITTAL INSTRUCTIONS FOR FORMAL REQUEST FOR SOFTWARE OR DATA

1. Date - Today’s date.
2. Title - Full application name or title of document (include the associated SC, LC, or CN numbers if known).
3. Originator - Your name.
4. ST - enter the task SC number followed by a ‘-’ followed by the task LC number, if known. (SCM will fill in the last unique identifying number.).
5. Organization - Your organization (i.e., EES-13).
6. Phone -Your work phone number.
7. Mail Stop - If for Los Alamos YMP use, your mail stop. Otherwise put N/A.
8. Check either the YMP USE (only for use to support the Yucca Mountain Site Characterization Project) or EXTERNAL USE box.

Instructions 9-14 are for the FILE TRANSMITTAL BLOCK.

9. Contact - Enter “ORIGINATOR” or name of person who will handle the transfer.
10. Phone - Contact work phone number.
11. Mail Stop - Contact mail stop, if not Originator and YMP use.
12. Hardware Platform - Destination platform (i.e., VAX, Sun SPARC, IBM compatible PC).
13. Operating System - Destination computing systems operating system (i.e., VAX VMS, UNIX, DOS, OS/2, IBM MVS).
14. File Format - Specify the preferred file format for transmission.

Instructions 15-16 are for the METHOD OF DISTRIBUTION BLOCK.

15. Method of Distribution - Specify the preferred method of transmission by checking one of the Tape, Disk, or Network File Transfer boxes.
16. Details - Provide the detailed information needed to make the transfer. (If for external use provide your full mailing, e-mail, or network address based on your transfer choice.).
17. Sign and date the ORIGINATOR line at the bottom of the form.

Los Alamos
Yucca Mountain Site Characterization Project

LIFE CYCLE

DATE: ____________ TITLE: ______________________________

DEVELOPER: ______________________________ **LC -** ______________

ORGANIZATION: ______________ PHONE: ____________ MAIL STOP: ________

☐ **REQUIREMENTS PHASE**		REVIEW	SEQUENCE
☐ REQUIREMENTS BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ SRS		☐ SRR	________
☐ **DESIGN PHASE**		REVIEW	SEQUENCE
☐ PRELIMINARY DESIGN BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ SDD ☐ PRIMARY MODULES			
☐ V&V PLAN BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ SDD			
☐ DETAILED DESIGN BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ SDD ☐ VVP (Plan Only) ☐ PRIMARY MODULES			
☐ MMS		☐ CDR	________
☐ **IMPLEMENTATION PHASE**		REVIEW	SEQUENCE
☐ V&V PROCEDURES BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ VVP ☐ SUPPORT MODULES			
☐ IMPLEMENTATION BASELINE		☐ IN-PROCESS	________
☐ FILE LIST ☐ VVP ☐ TEST RESULTS		☐ SAR	________
☐ USERS MANUAL ☐ MMS ☐ VDD			
☐ PRIMARY MODULES ☐ SUPPORT MODULES ☐ VVR			

CCB CHAIR: ______________ ______________ ________
Print name Signature Date

LANL-YMP-QP-03.21

Los Alamos
Yucca Mountain Site Characterization Project

LIFE CYCLE INSTRUCTIONS

1. Date - Enter today's date.
2. Title - Use the same title as found on the associated SC form.
3. Developer - CCB assigned developer name.
4. LC - number of the associated SC followed by a '-' followed by a unique sequence number. (SCM fills this in.).
5. Organization - CCB assigned developer organization (i.e., EES-13).
6. Phone - CCB assigned developer work phone number.
7. Mail Stop - CCB assigned developer mail stop.
8. Check the corresponding boxes.
9. The CCB Chair signs and date.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

1.0 INTRODUCTION (6.1)

1.1 Purpose (6.1.1)

This section specifies the purpose of the SRS and identifies its intended audience and use.

1.2 Scope (6.1.2)

This section identifies the software by name, explains what the software will do, and describes the application of the software being specified.

1.3 Definitions and Acronyms (6.1.3)

All terms and acronyms required to understand the SRS are defined in this section.

1.4 References (6.1.4)

All documents referenced elsewhere in the SRS are cited in this section.

2.0 GENERAL DESCRIPTION

2.1 Software Perspective (6.2.2)

This section describes the relationship of the software being specified to existing systems.

2.2 Software Functions (6.2.3)

This section describes the major functions that the software will perform.

2.3 User Characteristics (6.2.3)

Identify any constraints imposed by the user environment, such as the level of experience of potential users.

2.4 General Constraints (6.2.4)

Any other issues that constrain the developer's options for designing the system are identified here.

2.5 Assumptions and Dependencies (6.2.5)

List specific assumptions and dependencies (such as the hardware and software operating environment) here.

Include applicable software and hardware operation issues, to include programming languages and versions, portability, maintainability, reliability, and efficiency.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

3.0 FUNCTIONAL REQUIREMENTS

This section specifies the processes that transform the software inputs into outputs.

3.1 Major Function 1

3.1.1 Functional Requirement 1

3.1.1.1 Introduction

3.1.1.2 Inputs

3.1.1.3 Processing

3.1.1.4 Outputs

3.1.2 Functional Requirement 2

. . .

3.2 Major Function 2

4.0 EXTERNAL INTERFACE REQUIREMENTS (6.3.1.1)

This section specifies the interfaces that will be supported by the software.

4.1 User Interfaces (6.3.1.5.1)

This section specifies the software characteristics required to support each interface between software and user. It also specifies any requirements for optimizing these interfaces.

4.2 Hardware Interfaces (6.3.1.5.2)

This section specifies the logical characteristics of each interface between the software and the hardware environment within which it operates. It also specifies details of specific device support and required protocols.

4.3 Software Interfaces (6.3.1.5.3)

This section identifies other required software components with which the software is required to operate. It specifies the logical interfaces between these components and defines the requirements for any information interchange between them.

4.4 Communications Interfaces (6.3.1.5.4)

If interfaces to communications resources such as local area networks or data acquisition buses are required, they are specified here.

5.0 PERFORMANCE REQUIREMENTS (6.3.1.2)

Performance requirements such as number of users to be supported, number of files and records, sizes of tables and files, and transaction rates, if known, are specified in this section.

Los Alamos
Yucca Mountain Site Characterization Project

LANL-YMP-QP-03.21

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

6.0 DESIGN CONSTRAINTS

6.1 Standards Compliance (6.3.1.3.1)

Specify any special standards that constrain the design or implementation of the software. Routine organizational standards need not be explicitly identified in this section.

6.2 Hardware Limitations (6.3.1.3.2)

If hardware constraints exist for the operation of the software, they are enumerated in this section.

7.0 SECURITY (6.3.1.4.2)

Special requirements to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are specified here.

8.0 OTHER REQUIREMENTS

8.1 Data Base (6.3.1.6.1)

If a data base is to be developed as part of the project, specific requirements such as information content, frequency of use, data base access, organization, and data retention requirements are defined here.

8.2 Operations (6.3.1.6.2)

The requirements for routine or special operations required by the user such as operating modes, support functions, and backup and recovery operations are specified here.

SOFTWARE DESIGN DOCUMENT (SDD)

1.0 PURPOSE

A brief (one paragraph) description of the general requirements addressed by the software application is presented here.

2.0 FUNCTIONAL DESCRIPTION

This section provides a narrative description of the overall functional processing capabilities of the software application. A clear representation of the application's segmentation should be provided.

3.0 ASSUMPTIONS AND LIMITATIONS

Any assumptions and limitations inherent to the design of the software application are clearly stated in this section. List-format is recommended for presentation of these items.

4.0 PRIMARY PRODUCT DESCRIPTIONS

This section (and its subordinate subsections) provides detailed descriptions of each primary product (program or reuse component) associated with the application.

4.1 First Product

The name of the program or reuse component is specified here.

4.1.1 Purpose a description of the functional requirements satisfied by this product is presented here.

4.1.2 Functional Description Provide a description of the software structure, including software internal interfaces, control logic and data structure and flow.

4.1.3 Assumptions and Limitations Any assumptions and limitations inherent to the product are clearly stated in this section. List format is recommended for presentation of these items.

4.1.4 Input/OutputAll of the software product's input/output interfaces are presented in this section, organized into the subsections described below:

SOFTWARE DESIGN DOCUMENT (SDD)

4.1.4.1 Configuration Interface-Table

A representation of the product's configuration interface-table is provided in this section, if applicable. The name and description of each entry is described.

4.1.4.2 Other Input Data Files

Data elements that will be obtained from input files other than the product's configuration interface-table are described in this section. The name, structure, size, data type, bounds and description of each data element are provided.

4.1.4.3 Prompted Input

Each prompt issued by the product is specified in this section. The size, type, bounds, and any other restrictions on the allowed responses are also specified.

4.1.4.4 Output Files

A representation of each of the product's output file(s) is provided in this section. The name, structure, size, data type, bounds, and description of each data element are provided.

4.1.4.5 Displays and Reports

A representation of each display or report generated by the product is provided in a manner that specifies format and content.

4.1.4.6 Messages

The identifier and text of each message issued by the product is specified in this section. Conditions that are expected to result in the issuance of each message are described.

4.1.4.7 Other Input/Output

Any other formal input/output mechanisms (e.g. data bases, files, menus, command language, etc.) are described in detail in this section.

4.2 Second Product

. . .

SOURCE-CODE DOCUMENTATION-PROLOG

The following sections specify a functional outline for the source-code documentation-prolog that is required at the beginning of every created or modified source code module. The prolog is implemented as in-line comments using the source-program-language's comment-syntax and semantics.

The following sections describe the contents of each major section of the documentation prolog.

1.0 PURPOSE

This section provides a brief statement of the purpose of the source code module. It should not contain design information and should not be longer than one paragraph.

2.0 REVISION HISTORY

This section contains a tabular summary of the revision history of the module. Each line of the revision history specifies the date, author, authorizing Engineering Change Directive (change reference), and a brief (one sentence) description of the work performed. The table is organized in reverse chronological order, with the description of the most recent revision at the top of the table.

3.0 REQUIREMENTS TRACEABILITY

This section provides a list of the specific requirements, or general classes of requirements that are satisfied by the module. It is organized into a table that specifies the requirement number and name as recorded in the SRS.

4.0 SPECIAL COMMENTS AND REFERENCES

This section is used to document any special features of the module or its operating environment that are not appropriate for inclusion in one of the other sections of the prolog. The structure of special data objects may be included here, as may any other information that the author deems appropriate.

If a MMS is not required for the application under development, any major algorithmic equations will be discussed in this section. If the module makes use of information (algorithm, data, etc.) that should referenced, cite the source in this section.

The format of this section is left to the discretion of the author.

MODELS AND METHODS SUMMARY (MMS)

1.0 PURPOSE

The MMS provides a detailed description of the mathematical models and numerical methods employed by a software application.

2.0 DEFINITIONS AND ACRONYMS

Terms and acronyms that are used throughout the document are defined here.

3.0 REFERENCES

Supporting documents are cited here.

4.0 NOTATION

Identify all algebraic variables used in equations, supplying units when applicable.

5.0 STATEMENT AND DESCRIPTION OF THE PROBLEM

Describe the overall nature and purpose of the general analysis in which the model will be used.

6.0 STRUCTURE OF THE SYSTEM MODEL

Describe the role of any component models of the complete model.

7.0 GENERAL NUMERICAL PROCEDURE

Use flowcharts and block diagrams to describe the numerical solution strategy and computational sequence.

8.0 COMPONENT MODELS

8.1 Component Model 1

8.1.1 Purpose

Discuss the purpose and scope of the component model including input to and output from the model and how information is transformed by the model.

8.1.2 Assumptions and Limitations

Specify any assumptions and limitations upon which the model depends.

8.1.3 Derivation

Derive the model from generally accepted principles. Justify each step in the derivation. Clearly state the final mathematical form of the model.

MODELS AND METHODS SUMMARY (MMS)

8.1.4 Application

Discuss how the component model may be applied to a geologic repository. Describe any restrictions on the model, and state the validity of the model under extreme conditions.

8.1.5 Numerical Method Type

Characterize any numerical methods incorporated in the model that go beyond simple algebra.

8.1.6 Derivation of Numerical Model

Derive the numerical procedure from the mathematical component model, referencing all numerical methods. Present the final form of the numerical model and explain the algorithm.

8.1.7 Location

State where the component model is used in the software.

8.1.8 Numerical Stability and Accuracy

Discuss the stability and accuracy of the numerical model, and distinguish between those aspects that have been proven mathematically and those that have not been proven but have been observed in practice.

8.1.9 Alternatives

Discuss alternatives to the component model and state why this model was chosen.

8.2 Component Model 2

. . .

9.0 EXPERIENCE

Discuss the overall performance of the entire model.

V&V PLAN AND PROCEDURES (VVP)

The following sections describe the contents of each major section of the V&V Plan and Procedures.

1.0 PURPOSE

A brief (one paragraph) description of the goals of the V&V effort is presented here.

2.0 FUNCTIONAL DESCRIPTION

This section presents an overview of the organization and proposed execution of the V&V effort.

3.0 ASSUMPTIONS AND LIMITATIONS

Any assumptions and limitations inherent to the V&V effort are clearly stated in this section. Hardware and operating system environment will be specified in this section.

4.0 VERIFICATION AND VALIDATION PLAN

This section provides detailed descriptions of the V&V effort including the role of formal testing, and any other methods such as inspection, analysis, and demonstration. If the formal testing is organized into test suites, one subsection is included for each test suite.

4.1 Name of First Test Suite

4.1.1 Purpose

A brief (one paragraph) description of the purpose the test suite is presented here.

4.1.2 Functional Description

This section specifies, in narrative format, the overall organization of this test suite.

4.1.3 Assumptions and Limitations

Any assumptions and limitations inherent to the test suite are clearly stated in this section.

4.1.4 Summary of Test Cases

This section describes each test case in the test suite. It is organized into the following subsections:

4.1.4.1 Identifier for Test Case 1.

This section specifies a unique identifier for the test case.

a. Function Tested. This section provides a brief narrative description of the function under test.

V&V PLAN AND PROCEDURES (VVP)

b. Test Scope. This section specifies whether the test case is a verification, validation, regression, or acceptance (installation) test.

c. Requirements Tested. This section provides a list of the specific requirements (referenced by SRS paragraph number) that are verified by the test.

d. Required Inputs. This section describes the inputs necessary to configure the test.

e. Expected Outputs. This section defines specific criteria that will be met to pass the test.

4.1.4.2 Identifier for Test Case 2.

. . .

4.2 Name of Second Test Suite

. . .

USERS GUIDE FOR REUSE COMPONENTS (UG)

This Users Guide may be used in conjunction with reuse components. The following sections describe the contents of each major section of the Users Guide.

1.0 FORMAT

Show how the reuse component is invoked.

2.0 RETURNS

If the reuse component is a function, specify the data type and access mode (read, write, or read / write) for the returned value. If the reuse component is not a function, this section should be omitted from the Users Guide.

3.0 ARGUMENTS

List each formal parameter of the reuse component in this section. Specify the data type and access mode of each. Provide a brief narrative description of the use of the parameter.

4.0 DESCRIPTION

Describe the purpose and use of the reuse component.

5.0 RETURN STATUS

Describe all status codes and messages that are returned or issued by the reuse component.

6.0 ENVIRONMENTS

Enumerate and describe any external entities (such as include files) that are necessary to successfully compile the reuse component.

7.0 EXTERNALS

Enumerate and describe any external entities (such as libraries or library elements) that are necessary to successfully link the reuse component into a user application.

8.0 INSTALLATION

Describe the procedure that will be followed to install the reuse component on a user's system.

9.0 EXAMPLES

Provide examples of the use of the reuse component.

Los Alamos
Yucca Mountain Site Characterization Project

USERS MANUAL (UM)

The following sections describe the contents of each major section of the Users Manual.

1.0 PURPOSE

Specify the purpose of the Users Manual here.

2.0 DEFINITIONS AND ACRONYMS

Any terms or acronyms used throughout the document are defined in this section.

3.0 REFERENCES

Any documents referenced within the Users Manual are cited here.

4.0 PROGRAM CONSIDERATIONS

4.1 Program Options

Each major program option (and combinations of options) is described in this section; to include input and output options.

4.2 Initialization

Initialization values will be specifically identified in this section.

4.3 Restart Procedures

Describe the restart capabilities of the software.

4.4 Error Processing

Describe error messages, error paths, and abnormal termination sequences in this section. Include anticipated errors and how the user can respond.

5.0 DATA FILES

5.1 Name of First Data File

5.1.1 Content

Describe the content, purpose, and organization of the data file.

5.1.2 Use by Program

Specify whether the data file is used for input or output, describe the role of the information in the data file, and identify at what stage during program execution it is used.

USERS MANUAL (UM)

5.1.3 Auxiliary Processing

Identify any auxiliary programs used to create, modify, reformat, or interpret the data in the data file.

5.2 Name of Second Data File

. . .

6.0 INPUT DATA

6.1 General Considerations

6.1.1 Techniques

Any special input techniques or requirements are described in this section.

6.1.2 Consecutive Cases

Give conditions for retention of input data between cases, if any.

6.1.3 Defaults

Explain any conventions regarding default values.

6.2 Individual Input Records or Parameters

The format outlined below may be replaced by a table or tables containing the information specified. If any of the topics specified can be defined globally for all variables, identify the topic and its entry here.

6.2.1 Record / Parameter 1

6.2.1.1 Record Identifier

Provide the line identifier or data object label for this type of record.

6.2.1.2 Input Variables

Identify the internal variables that are assigned values from data provided in this record. If this is the same as the Record Identifier in all cases, omit this section.

6.2.1.3 Format

Specify the format of the record, if any.

6.2.1.4 Need

Specify whether input is necessary or optional for each variable.

6.2.1.5 Repetition

Discuss how many of these records are required or may be used optionally.

Los Alamos
Yucca Mountain Site Characterization Project

LANL-YMP-QP-03.21

USERS MANUAL (UM)

6.2.1.6 Units

Provide the dimensional units for each input field.

6.2.1.7 Default

Provide the default value, if any, for each field.

6.2.1.8 Description

Define the meaning of each variable and discuss its use within the code.

6.2.1.9 Range

State the acceptable limits for each variable.

6.2.2 Record / Parameter 2

. . .

7.0 OUTPUT

Discuss the code output with respect to input options, and state the origin and meaning of output variables, including a description of the allowable and tolerable ranges. Describe any graphical capabilities of the software.

8.0 SYSTEM INTERFACE

8.1 System Dependent Features

Identify the external references that will be supplied by the system.

8.2 Compiler Requirements

Identify the compilers that have been successfully used, and specify any special load options.

8.3 Hardware Requirements

Describe any special hardware features or environments required by the software.

8.4 Software Environment

Describe software dependencies of this software. Examples are run-time libraries, graphic processors, and other pre or post processors.

USERS MANUAL (UM)

8.5 Control Sequences or Command Files

Describe any special command sequences required by the software.

8.6 Installation Instructions

Provide a comprehensive set of instructions for installing the software on platforms for which it has been designed. In a separate command file or script, provide an executable procedure for compiling and/or linking the application. Reference the name of the file here.

9.0 EXAMPLES AND SAMPLE PROBLEMS

Furnish a suite of sample problems that demonstrate how the software is used. Include all input data and sample output, either as file listings or by referencing accompanying files.

10.0 USER SUPPORT

Specify that users should contact the Los Alamos SCM for user and maintenance support.

VERSION DESCRIPTION DOCUMENT (VDD)

The following sections describe the contents of each major section of the VDD.

1.0 VERSION IDENTIFICATION

This section specifies the release identifier for the application or data described by this document (supplied by SCM).

2.0 REFERENCES

Any documents referenced in the VDD are cited here.

3.0 CHANGE SUMMARY

3.1 Release Notes

This section contains a summary of the revision history of the application, and presents a description of the new features of the application. In addition, this section identifies any compatibility issues with respect to previous versions and identifies any known errors that are present in the released version.

3.2 SCs Satisfied

This section contains final copies of (or references to) any SC forms that are satisfied by the release of this version of the software (supplied by SCM).

4.0 CHRONOLOGICAL RECORD OF DEVELOPMENT

This section is a compendium (beginning with the authorizing SC) of all of the supplemental documentation forms that were generated in support of the development of this version (supplied by SCM).

This section may be omitted from the VDD that is distributed to users of the software.

V & V REPORT (VVR)

The following sections describe the contents of each major section of the V&V Report.

1.0 SCOPE OF V&V ACTIVITIES

Describe the overall V&V effort and the V&V philosophy. Specify why they are appropriate.

2.0 REFERENCES

Cite any references to standard works here.

3.0 DESCRIPTION OF ENVIRONMENT

Summarize the hardware and software characteristics of the test environment here.

4.0 DISCUSSION OF TEST RESULTS

Discuss and interpret the TR in this section. Particular emphasis is placed upon discrepancies between the actual and expected TR. An analysis of any such discrepancies (as well as an explanation of any mitigating circumstances) is presented in this section.

In the event that no discrepancies are identified for a test suite, that fact may be noted and no further explanation is required.

4.1 Test Suite 1

4.2 Test Suite 2

. . .

5.0 ADDITIONAL V&V ISSUES

Discuss any aspects of the V&V effort that are outside of the scope of the formal testing effort.

6.0 CONCLUSIONS AND RECOMMENDATIONS

This section contains the author's conclusions regarding the acceptability of the V&V effort. The author may also make recommendations regarding the disposition of discrepancies discovered during the testing program. For example, the author may recommend that a particular discrepancy remains not addressed pending a future repair operation. The recommendation would require that the error be documented in the VDD and that the application be accepted with the known defect. Recommendations may also be made regarding modified or enhanced test strategies that should be applied to future versions of the application.

Los Alamos
Yucca Mountain Site
Characterization Project